AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions and listings of claims in the application:

1. (Currently amended) A method for developing an application, the method comprising:

defining file borders for development objects in a data model, wherein the data model includes a component class and a model class associated with the component class, and a controller class, associated with the component class, that associates a user interface to a business application model;

storing the development objects of the application in a file-based repository based on the file borders; and

employing an API derived from the data model to access the development objects.

2. (Original) The method of claim 1 further comprising caching the development objects in a local cache.

3. (Original) The method of claim 1 wherein defining the file borders comprises identifying one of the development objects as a main development object to be included in a file with any development objects that are defined in the data model to be children objects of the main development object that are not identified as main development objects.

4. (Original) The method of claim 3 further comprising storing in the file user-defined code associated with the main development object.

5. (Original) The method of claim 3 further comprising storing in the file a reference to another development object stored in another file.

6. (Original) The method of claim 1 further comprising enabling a user to define a source path for one of the development objects.

7. (Original) The method of claim 1 wherein employing the API further comprises using tools that use the API to enable a user to perform a development operation.

8. (Original) The method of claim 7 wherein the development operation includes a copy and paste operation.

9. (Original) The method of claim 7 wherein the development operation includes enabling a user to refactor a copied development object.

10. (Original) The method of claim 9 further comprising enabling a user to define a scope of the refactor.

11. (Original) The method of claim 7 wherein the development operation includes storing translatable text separate from the development objects.

12. (Currently amended) A method for developing applications, the method comprising:

generating a data model for an application, the data model being implemented in a language that includes ~~an~~ customizable extension, the data model including a feature defined using the customizable extension, wherein the data model includes a component class and a model class associated with the component class, and a controller class, associated with the component class, that associates a user interface to a business application model;;

deriving an API from the data model, the API incorporating the feature; and

enforcing constraints specified in the data model by employing the derived API during development of the application.

13. (Original) The method of claim 12, wherein the feature comprises an indication used to implement a file border.

14. (Original) The method of claim 12, wherein the feature comprises an indication used to implement a platform-specific feature.

15. (Original) The method of claim 12, wherein the feature comprises an indication representing translatable text.

16. (Original) The method of claim 12, wherein the feature comprises an indication representing that an aggregation in the data model is ordered.

17. (Original) The method of claim 12, wherein the feature comprises an indication representing a singular name.

18. (Original) The method of claim 12, wherein the feature comprises an indication representing that an attribute in the data model is nullable.

19. (Currently amended) A computer program product containing instructions which, when executed on a processor, form a system for developing an application, the system comprising:

a repository storing development objects using file borders defined in a data model, wherein the data model includes a component class and a model class associated with the component class, and a controller class, associated with the component class, that associates a user interface to a business application model;;

a local development cache for caching the development objects from the repository;

an API derived from the data model; and

a user interface development tool that uses the API to access the development objects.

20. (Original) The system of claim 19, further comprising a repository server that includes the repository.

21. (Original) The system of claim 19, wherein the user interface development tool comprises one of a project browser, an application modeler, a view designer, a controller and context editor, and a model editor.